

軟性計算理論與知識處理技術之研究

淡江大學 資圖系
助理教授 林信成

On Soft Computing Theory and Knowledge Processing Technology

Sinn-Cheng Lin

Assistant Professor
Department of Information and Library Science,
Tamkang University,
Taipei, Taiwan, R.O.C.
E-Mail: sclin@mail.tku.edu.tw

ABSTRACT

Knowledge is the essential element of human thoughts. Creating an artificial system with knowledge and intelligence, furthermore, has been one of the humankind's long-term visions. The computer has been the smartest machine ever since human history in technology began. Therefore, people attempted to use computer to realize human intelligence such as thinking, reasoning, learning. However, it always encountered numerous bottlenecks. The operating model of a computer is absolutely different from that in human brain. The computer is superior to human brain by its computing ability. But the computer loses its efficiency in conceptualizing, thinking, reasoning and distinguishing. The soft computing theory is developed to make a connection between machine intelligence and human brain. With such a theory, researchers on knowledge engineering can transfer the minding process, recognition structure and learning procedure of human into mathematical models. Then, they could work on any kinds of analysis, designing, simulation and experiments systematically. By doing so, the knowledge processing ability of artificial intelligent system can be improved and enhanced.

This paper is focused on the integration issues of soft computing theory. It discusses the application potential of soft computing in the fields of machine thinking, machine recognizing, and machine learning. In addition, it also anatomizes the role of knowledge processing technologies that based on soft computing theory play in modern knowledge engineering.

一、緒論

「知識便是力量」乃亙古不變的真理。隨著資訊科技的發展，知識管理（Knowledge management）成為近年來頗為熱門的研究主題，知識經濟（Knowledge economics）更成了政府的施政方針。圖書館是知識的蘊藏寶庫，也是人類智慧的匯聚場所，因此，對於與知識有關的研究課題，圖書館界自然而

然感到興趣盎然。然而，知識是什麼？如何形成？其本質又為何？該如何表達？又該如何處理？這些都是人們在研究有關知識的主題時所面臨的根本問題。

知識（Knowledge）和智慧（Intelligence）是一體的兩面。簡單的說，知識是智慧的結晶，更是人類文明的重要資產。人類經由認知、學習、思考等心智活動，以獲取、創造、累積知識，並藉由各種媒體，儲存、傳播、共享知識。長久以來包括醫學、工程學、物理學、生物學、心理學、人文社會科學等各個領域的研究者，無不投注心力從不同的角度、不同的觀點來研究及解釋人類的心智活動，試圖揭開人類智慧的神秘面紗。電腦的發明使得人們開始研究使用機器協助人類進行智慧活動的可行性，促使人工智慧（Artificial intelligence，簡稱 AI）自 1950 年代起逐漸成為一個重要的研究領域，希望有朝一日能創造出如同人類一樣具有思維能力的機器。人工智慧所要解答的首要問題便是「人類是如何進行智慧活動的？」這真是大哉問！從最根本的感覺、知覺、學習、記憶，到較高階的思考、推理、決策、語言，以至更複雜的社會行為、人格差異等，無一不是心智活動的一部份。由於問題龐大複雜，在人工智慧發展多年之後，人們終於發現要建構一部能像人類一樣解決眾多問題的通用型機器，簡直遙不可及。因此，人工智慧領域便開始分支出許多針對單一主題的研究，包括了知識工程（Knowledge engineering）、專家系統（Expert systems）、自然語言處理（Natural language processing）、電腦視覺（Computer vision）、機器人學（Robotics）、機器學習（Machine learning）... 等，以及近年來逐漸整合成形的軟性計算（Soft computing）。

知識工程是一門研究以數理方法擷取、表達、處理知識的科學與技術，乃為建構智慧型系統的理論基礎。數理方法經常將複雜問題加以簡化以利分析研究，如物體運動可簡化為牛頓運動定律、電磁現象可簡化為馬克斯威爾方程式等。知識工程將複雜的人類思維，視為一連串藉由各種管道獲取知識（Knowledge），然後進行推理（Reasoning），最後做出決策（Decision）的過程，由此可見「知識」乃是人類智慧中最基本的一環。知識的獲得需要經過感知、學習、記憶才能逐漸累積起來；有了足夠的知識，才能做出正確的推理與決策，此乃人類思維的基本體系。至於軟性計算則是近年來，綜合許多不同領域的研究，如模糊理論、類神經網路、學習演算法 ... 等，所發展出來的整合性知識工程技術。長久以來人們雖然一直試圖利用電腦來模擬人類的知識思維，卻經常遭遇瓶頸，究其原因，電腦強於人腦之處在於計算能力，對於無法以有效的計算法則解決的問題，如概念、思考、推理、識別 ... 等，電腦就略遜一籌了。軟性計算理論最主要的功能在於將人類心智活動、認知結構加以模式化、數學化，使得知識工程研究者能進行有系統的分析、設計、模擬、實驗，以便在機器智能與人類智慧間建立起一座連結之橋，進而提昇智慧型系統對於知識處理之能力。

本文從軟性計算理論整合的趨勢著眼，探討其在機器思維、機器認知、機器學習等方面所展現的應用潛力，並剖析以軟性計算理論為根基的知識處理技術在當代知識工程中所扮演的角色。

二、知識工程與知識處理

「知識」是人類思維的基本要素，因此，要讓機器具備智慧進而如同人類一樣理解事物，就必須研究以機器擷取、表達、處理知識的技術與方法，也就是所謂的知識工程，專家系統（Expert system）便是知識工程最成功的應用。專家系統不僅是一個知識管理系統，更是一個具有模擬人類專家決策能力的系統¹。一

¹ Giarratano and Riley, Expert Systems: Principle and Programming, PWS-KENT, 1989, Page 1.

個典型的專家系統，必須利用各種知識擷取 (Knowledge acquisition) 管道獲取專家的領域知識 (Domain knowledge) 再以知識表示法 (Knowledge representation) 將人類知識轉存成知識庫 (Knowledge base)，而在知識庫之上有一個具備邏輯推理能力的推論引擎 (Inference engine)，使用者經由使用者介面輸入欲查詢的事實 (Facts)，推論引擎將該事實與知識庫中的先備知識進行比照，做出建議性的推理和決策，如圖 1 所示。所以專家系統可視為是「一個以知識為依據，以推理為方法的智慧型系統」²。依此觀之，「知識」和「推理」不但是人類智慧的結晶，更是專家系統中不可或缺的核心要素。

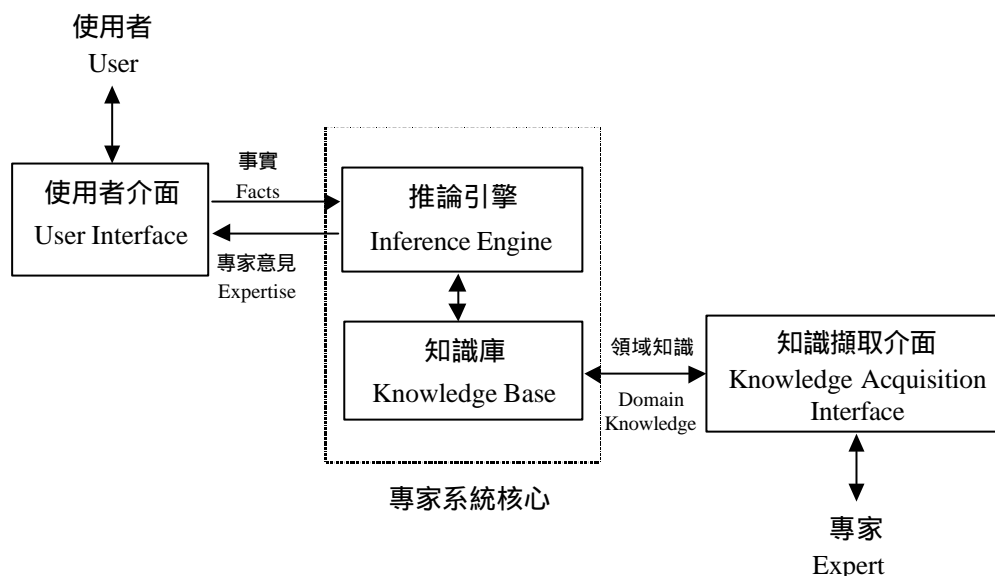


圖 1 專家系統概念圖

知識表示法是表達知識的基本方法，乃知識工程中最基本的研究主題之一。有了知識表示法，「知識」才能被轉化為電腦能理解的數理模型，進行自動化處理。目前，已被提出的知識表示法眾多，較重要的有：規則式 (Production rules) 表示法、邏輯與集合 (Logic and sets) 表示法、語意網路 (Semantic nets) 表示法、框架 (Frames) 表示法、劇本 (Scripts) 表示法、物件導向表示法 (Object-oriented representation) 等³。

傳統上，知識工程或知識表示法皆採用古典的數理方法來運作，如古典邏輯 (Classical logic)、古典集合論 (Classical set theory) ...等，其特色是凡事講求黑白分明，馬虎不得。這樣的數理方法稱為硬性計算法 (Hard computing)，我們已經見到此種方法應用在以數位式方式運作的當代電腦上，作為資料處理、分析的工具時，具有無比的威力並獲得了空前的成功；但是，當電腦的用途要再往上提昇到知識處理的層次時，卻困難重重、屢遭瓶頸，於是造就了軟性計算的興起。

三、軟性計算與機器智能

軟性計算並不是一門單獨發展的理論或技術，而是綜合許多不同領域的研究，所發展出來的整合性知識工程技術，如模糊邏輯、類神經網路、遺傳演算法 ...

² 葉怡成、郭耀煌，專家系統—方法、應用與實作，民 80，全欣資訊圖書，頁 7。

³ 同註1，頁 63-102。

等。軟性計算理論最主要功能在於將人類心智活動、認知結構加以模式化、數學化，使得知識工程研究者能進行有系統的分析、設計、模擬、實驗，例如吾人可以類神經網路模擬人腦之感知及認知結構，以學習演算法進行學習過程，以模糊理論完成思考、推理與決策。如此，便可在機器智能與人類智慧間建立起一座連結之橋，進而提昇智慧型系統對於知識處理之能力，也就是所謂的「機器智商」（Machine Intelligence Quotient，簡稱 MIQ）。

軟性計算不同於傳統硬性計算（Hard computing）的地方在於硬性計算講究真假分明，非 0 即 1；而軟性計算卻容忍非精確性（Imprecision）、不確定性（Uncertainty）部份真實性（Partial truth）的存在^{4, 5}，更符合人腦的思考模式，有利於知識的自動化處理。

（一）模糊推論與機器思維

當代數位電腦採用布林邏輯（Boolean logic）二值法的硬性計算模式運作，雖然在資料處理層面獲得了重大的成就，然而在知識處理層面卻一直碰到困難及瓶頸。這是因為在真實世界中，人類有許多思維過程是非常「模糊」的，例如：「如果路況很不好，則車速慢一點」，其中「很不好」、「慢一點」都是沒有明確定義的模糊概念，但是人腦就是能夠處理，實在神奇，這與需要精確數據才能計算的電腦，在運作上顯然有著極大的差異。想一想，人腦並未經過詳細精確的運算，卻能做出正確的判斷，電腦卻辦不到，為什麼？因為「數值計算並不同於知識思維」！事實上，當人們在做決策時，經常是使用一些模糊語意規則，諸如：「如果 A ，則 B 」，但一直以來都沒有足以描述人類思維這種模糊性的數學方法，以供電腦遵循，直到 L. A. Zadeh 於 1965 年提出了模糊集合論（Fuzzy sets）⁶和 1972 年的近似推理（Approximation reasoning）⁷之後，總算有了一個數學理論可以處理人類思維的語意資訊。「Fuzzy」一詞在字典中的解釋是「似毛絨的」，引申為「模糊的」、「朦朧的」之意，英文上的含意是負面的，因此起初並未廣被一切講求精確的工程研究人員所接受。直到 70 年代初期出現成功的工業應用⁸，才逐漸受到關注，但仍停留在學術研究階段；80 年代末期，日本結合各界力量，大舉開發 Fuzzy 家電產品，才將其推向市場，引起廣泛注意。90 年代起國內外的產、官、學界紛紛投入大量的人力物力進行模糊理論之研究，各式各樣的應用也紛紛出爐，印證了模糊理論的實用性。如今，模糊理論已經成為一門專門探討人腦如何利用模糊訊息或不完全資料，不需經過精密繁雜的計算過程，仍能作出正確判斷，進而瞭解人腦是如何從模糊作決策的理論⁹，其涵蓋範圍極為廣泛，包括模糊集合（Fuzzy set）、模糊關係（Fuzzy relation）、模糊邏輯（Fuzzy logic）、模糊量測（Fuzzy measure）、模糊推理（Fuzzy reasoning）...等¹⁰；應用領域更是包含了模糊控制（Fuzzy control）、模糊決策（Fuzzy decision）、模糊專家系統（Fuzzy

⁴ L. A. Zadeh, "What is soft computing?" BISC homepage, 1994. Available at <http://www.cs.berkeley.edu/projects/Bisc/bisc.memo.html#what_is_sc>.

⁵ X. Li and D. Ruan, "Discussion on Soft Computing at FLINS'96," International Journal of Intelligent Systems, vol. 13, 287-300 (1998).

⁶ L. A. Zadeh, "Fuzzy Sets," Information and Control, Vol. 8, pp. 338-353, 1965.

⁷ L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," IEEE Trans. on Systems, Man and Cybernetics, SMC-3, no. 1, pp.28-44, 1973.

⁸ E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," Proc. IEE, vol. 121, no. 12, pp. 1585-1588, 1974.

⁹ 林信成、彭啟峰，「Oh! Fuzzy — 模糊理論剖析」，第三波，民 83。

¹⁰ G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall: New Jersey, 1988.

Expert system) 模糊資訊檢索(Fuzzy information retrieval) 模糊影像處理(Fuzzy Image processing) 模糊模式識別 (Fuzzy pattern recognition) 模糊規劃 (Fuzzy programming) 模糊資料庫 (Fuzzy database) ... 等¹¹。

模糊理論的發展使得人類思維與電腦運算之間搭起了一座交流的橋樑,它可以將模糊概念加以量化,例如日常用語中許多曖昧不明、含意模糊的語句:昨天「很熱」、你的車速「太快」了、他並「不太老」... ,其中「很熱」、「太快」、「不太老」並未指明真正的數值是多少,但人腦卻可以很容易的接收並處理,這些模糊語意詞(Fuzzy linguistic terms)經由模糊集合的歸屬函數(Membership function)加以量化後,便可交由電腦處理,而電腦也因此而稍具人腦思考的雛形。

以下說明在模糊理論中幾個重要的概念。

1. 模糊集合

在論域 (Universe of discourse) X 中的任一模糊集合 A , 可由以下的歸屬函數加以描述 :

$$m_A: X \rightarrow [0, 1] \quad (1)$$

其中 $x \in X$, $m_A(x)$ 代表 x 隸屬於 A 的歸屬度。

2. 模糊集合運算

令 A 和 B 分別為論域 X 中的兩個模糊集合, 則

(1) 交集運算 (Intersection)

A 和 B 的交集記作 $A \cap B$, 亦為 X 中的一個模糊集合, 其歸屬函數為 :

$$m_{A \cap B}(x) = T(m_A(x), m_B(x)) \quad (2)$$

其中 $T(\cdot)$ 稱為 T-範數 (T-norm) , 是一個從 $[0, 1] \times [0, 1]$ 映射至 $[0, 1]$ 的函數。常用的幾個 T-norm 函數為 :

$$\text{模糊交集 (fuzzy intersection) : } T(a, b) = \min(a, b) \quad (3a)$$

$$\text{代數乘積 (algebraic product) : } T(a, b) = ab \quad (3b)$$

$$\text{界限乘積 (bounded product) : } T(a, b) = \max(0, x + y - 1) \quad (3c)$$

(2) 聯集運算 (Union)

A 和 B 的聯集記作 $A \cup B$, 亦為 X 中的一個模糊集合, 其歸屬函數為 :

$$m_{A \cup B}(x) = C(m_A(x), m_B(x)) \quad (4)$$

其中 $C(\cdot)$ 稱為 T-共範數(T-conorm) , 是一個從 $[0, 1] \times [0, 1]$ 映射至 $[0, 1]$ 的函數。常用的幾個 T-conorm 函數為 :

$$\text{模糊聯集 (fuzzy union) : } C(a, b) = \max(a, b) \quad (4a)$$

$$\text{代數和 (algebraic sum :) : } C(a, b) = a + b - ab \quad (4b)$$

$$\text{界限和 (bounded sum) : } C(a, b) = \min(1, a + b) \quad (4c)$$

¹¹ 同註9, 頁 1-7~頁 1-16。

(3) 補集運算 (Complement)

A 的補集記作 \bar{A} ，亦為 X 中的一個模糊集合，其歸屬函數為：

$$m_{\bar{A}}(x) = 1 - m_A(x) \quad (5)$$

3. 模糊關係

模糊關係為明確關係 (Crisp relation) 的擴展。明確關係簡稱關係 (Relation) 是用以描述事物之間關聯性的一種數學表示法，例如 x 與 y 兩個人是兄弟，我們就說 x 和 y 具有「兄弟關係」，從數學的觀點來看，可以表示成一個「二元關係」 (Binary Relation) R ，記為 xRy 。如果將這種觀念從事物與事物之間擴展到集合與集合之間，而將 x 與 y 分別視為集合 X 與集合 Y 的元素，那麼以上所說的這個 R 即可表為定義於 $X \times Y$ 的部份集合 $R(X, Y)$ ，記作 $R(X, Y) \in X \times Y$ 。二元關係可以很簡單的利用關係矩陣來描述，只要將集合 X 的元素作為列標 (Row index)，集合 Y 的元素作為行標 (Column index)，並將各組配對對應於關係 R 的特性函數值 (無關係為 0，有關係為 1) 填入適當位置即可。

在明確關係的定義中，事物之間的關係只有「有關係」及「無關係」兩種情況，並無強弱好壞之區別。但是真實世界中並非如此單純，以文件分類為例，現有 A, B 兩份文件，如果以明確關係來表示，只能說彼此「有」關係 (同類) 或「沒有」關係 (不同類)，但是並無法瞭解其關連程度如何，這是一種硬性分類 (Hard classification)。所謂「模糊關係」就是將明確關係的觀念推廣，以歸屬函數作為關係的強弱指標，當兩者的關係程度愈強，其歸屬度便愈接近於 1；反之，便愈接近於 0。所以，當事物之間的關係不明確或有強弱之分時，亦即無法單純的用「有」關係或「無」關係來表示時，模糊關係便成了相當有用的工具，於是便可完成諸如「文件 A 與 B 的關連度要比與文件 A 與 C 的關連度大一些」的軟性分類 (Soft classification)。從集合的觀點而言，集合 X 與集合 Y 之間的模糊關係 $R(X, Y)$ ，可視為定義於 $X \times Y$ 上的模糊集合，亦即 $m_R: X \times Y \rightarrow [0, 1]$ ，其中， m_R 為關係 R 的歸屬函數，用以描述兩者關係程度的強弱。

近年來模糊關係在網路上應用得相當普及，例如許多搜尋引擎在進行全文檢索時，所提供的模糊檢索就是採用模糊關係作為判定文件與檢索詞相匹配程度的評估指標。

4. 模糊邏輯與模糊推理

邏輯是一門古老的學問，早在古希臘時代，哲學家們早已普遍採用二值法進行推論，因此，古典邏輯又稱為二值邏輯 (Binary logic)。布林代數 (Boolean algebra) 和電腦的出現使二值邏輯的發展達到了巔峰，而且成為知識工程中極重要的知識表示法之一。古典邏輯中最常被提及的是命題邏輯 (Proposition logic)。所謂「命題」 (Propositions) 是可以判斷真 (True) 假 (False) 的一個「句式」 (Formula)，如「圖書館關了」、「那本書被借走了」、「這個學生不及格」... 等。透過邏輯運算子且 (AND) 或 (OR) 非 (NOT)，可以將簡易命題組成複合命題 (Compound propositions)。命題邏輯最基本的推論就是求解命題「若 P 則 Q 」的真值，其結果只有真或假兩種狀況，不能也不會有其他值。

模糊邏輯的命題則無法以真、假二分法來斷定，例如「這位老師很年輕」、「她很美」、「現在有點冷」... 等，因為在這些命題中的語意詞「很年輕」、「很美」、「有點冷」都是見仁見智的模糊語意詞，有些人覺得這個命題對，有人覺得不對，還有人覺得似是而非、半對半錯，像這樣以模糊語意詞表達命題的真值或可能性

值的方式，就是所謂模糊邏輯，我們可以用真、有點真、非常真、假、有點假、非常假等語意變數來描述一個命題的真實程度。模糊推理就是求解模糊命題「若 P 則 Q」的模糊真值，亦即真實程度，更契合人腦的運作方式。

5. 模糊推論系統

在眾多知識表示法中，「規則式表示法」無疑的是最符合人類思維模式，也是應用最廣的一種，其最主要的特色是將人類知識描述成一條一條「如果...則...」的推論規則，以供推論引擎使用。有別於古典邏輯採用精確推理，基於模糊邏輯所建構的模糊推論系統（Fuzzy inference systems）採用近似推理（Approximation reasoning），是一種擬人式的推論方式，以開車為例：人類在駕駛時，並未實際測量道路彎度、車與路口路邊的距離，也從未考慮過精確的方向盤角度、油門或剎車的大小，只憑藉一些模糊的訊息及規則，像是「如果道路有點右彎，則方向盤向右打一點並開慢一點」、「如果距路口很遠且路況良好，則開快一點」，即能將汽車控制於正常車道上。人腦並未經過詳細精確的運算，卻能做出正確的判斷，模糊推論系統就是被發展出來模擬人類這種「過程模糊，結果精確」的推論方式。

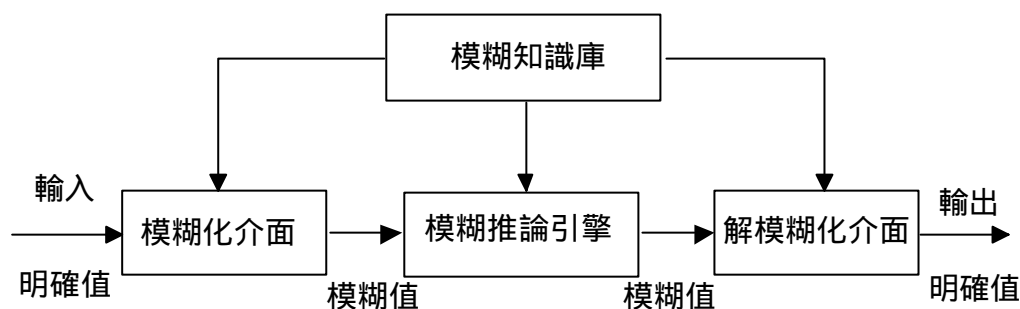


圖 2 模糊推論系統架構圖

一個典型的模糊推論系統如圖 2 所示，主要由四大功能模組所構成：模糊知識庫（Fuzzy knowledge base）、模糊推論引擎（Fuzzy inference engine）、模糊化介面（Fuzzifier）和解模糊化介面（Defuzzifier）。分述如下：

(1) 模糊知識庫

模糊知識庫是模糊推論系統中用以儲存人類知識的寶庫，主要由兩部份構成，其一為記錄專家經驗法則之模糊規則庫（Rule base），另一為定義模糊語意之資料庫（Data base）。模糊規則通常是以「若 ~ 則 ~」（「IF ~ THEN ~」）的方式記錄，例如老師在上課時，心中可能存在著一個教學指導原則：「如果學生程度好而且學習能力強，那麼課程進度就可以快一點而且課程難度也可以高一點」，寫成典型的模糊規則便是：

IF 學生程度「好」 AND 學習能力「強」
THEN 課程進度「快一點」 AND 課程難度「高一點」

其中，「好」、「強」、「快一點」、「高一點」等語意詞，都被視為模糊集合，其對應的歸屬函數必須在語意資料庫中加以定義。

若以通式表示，一個模糊規則庫可記作 $RB := \bigcup_{j=1}^N R_j$ ，是由 N 條模糊規則所組成，而每條規則 R_j 可表示為：

$$R_j : IF x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{ and } \dots x_n \text{ is } A_{nj} \\ THEN y \text{ is } B_j \quad (6)$$

其中 x_i ($i = 1, 2, \dots, n$) 為輸入變數, y 為輸出變數; 輸入語意值 A_{ij} 和輸出語意值 B_j ($j = 1, 2, \dots, N$) 分別是定義於輸入論域 X_i 和輸出論域 Y 的模糊集合。

(2) 模糊推論引擎

模糊推論引擎主要以近似推理為方法, 根據知識庫中的先備知識進行模糊推論得出結果。近似推理衍生自模糊邏輯, 其主要的邏輯運算雖然也使用 AND、OR、NOT ... 等, 但相較於布林邏輯二值運算 (Binary computing) 的結果只有 1 (邏輯真值 True) 0 (邏輯假值 False) 兩種情況, 模糊邏輯是一種軟性計算方式, 其結果介於 0~1 之間, 代表著不同等級的真假程度, 更符合人類思維的特性。

模糊推論過程如下: 將模糊規則的前件部各項輸入值與語意值進行模糊匹配, 轉換成模糊集合後, 規則庫中有關的模糊邏輯運算 AND、OR、NOT, 便等效於模糊集合的交集、聯集、補集運算了, 所得的結果稱為該條規則的觸發強度 (Firing strength):

$$m_{R_j}(\underline{x}) = \bigcap_{i=1}^n m_{A_{ij}}(x_i) \quad (7)$$

其中, $m_{R_j}(\underline{x})$ 代表第 j 條規則的觸發強度 ($\underline{x} = [x_1 \ x_2 \ \dots \ x_n]^T$), 交集符號 \cap 代表 AND 運算, 可用 T-norm 完成。經由觸發強度可以得出第 j 個模糊基底函數 (Fuzzy basis function) $r_j(\underline{x})$:

$$r_j(\underline{x}) = \frac{m_{R_j}(\underline{x})}{\sum_{j=1}^N m_{R_j}(\underline{x})} \quad (8)$$

(3) 模糊化介面

用以將明確輸入值 $x_0 \in X$ 轉換成定義於 X 的模糊集合 A_x 。特殊地, 若 A_x 是一個支點 (Support) 在 x_0 的模糊單點集 (Fuzzy singleton), 則稱此種模糊化方式為單點集模糊化 (Singleton fuzzification), 是一種最常見的模糊化方式。

(4) 解模糊化介面

相對於模糊化的反向動作, 解模糊化主要將定義於 Y 的模糊集合 B 轉換為精確的輸出值 $y \in Y$ 。最常用的解模糊化運算為「加權平均解模糊化法」(Weighted average defuzzification):

$$y = \frac{\sum_{j=1}^N j_j m_{R_j}(\underline{x})}{\sum_{j=1}^N m_{R_j}(\underline{x})} := \underline{j}^T \underline{r}(\underline{x}) \quad (9)$$

其中, $\underline{r} = [r_1 \ r_2 \ \dots \ r_N]^T$ 為模糊基底函數向量, $\underline{j} = [j_1 \ j_2 \ \dots \ j_N]^T$, j_j 為

輸出語意值 B_j 的支點。

模糊推論系統若再加上模糊知識擷取機制 (Fuzzy knowledge acquisition facility) 和使用者的介面 (User interface), 則成為模糊專家系統, 如圖 3 所示, 此二模組分述如下:

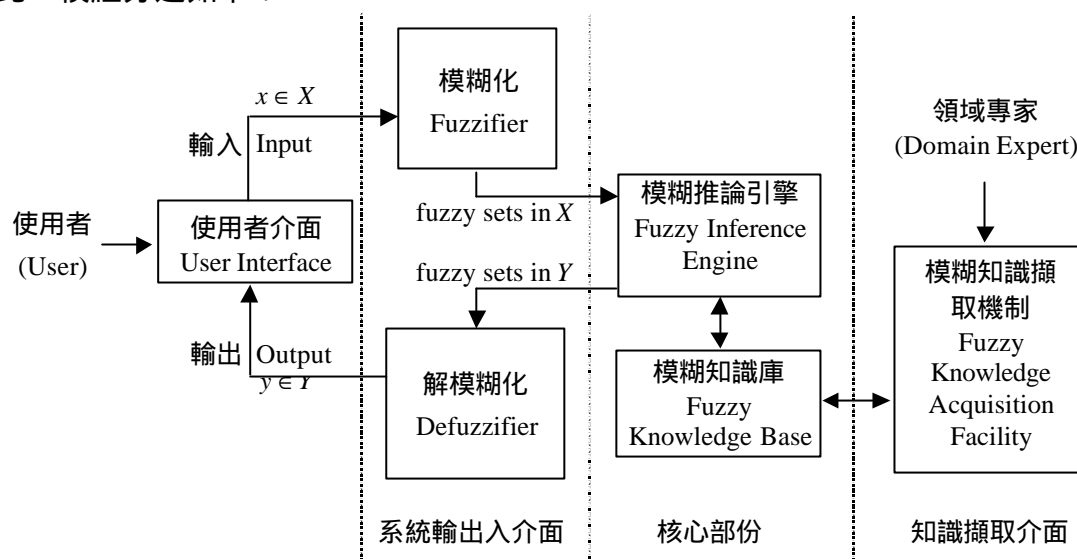


圖 3 模糊專家系統

(1) 模糊知識擷取機制:

知識庫中的知識仍需得至專家提供, 因此需要一個知識擷取模組來完成。模糊理論發展之初, 知識庫的建立大都倚賴領域專家 (Domain experts) 提供經驗, 再由知識工程師將其轉換成對應的模糊規則及歸屬函數。其優點是設計原理簡單且直觀; 而缺點則是費時耗力。因為知識的擷取並非易事, 且隨著所要解決的問題愈趨繁瑣, 推論規則也相對愈趨複雜而更難於歸納整理。幸好拜科技之賜, 今日的電腦運算能力相較於十年前, 可謂一日千里, 加以眾多學習理論日趨成熟, 為模糊推論系統的建立另外提供了出路。目前已有許多研究成果朝著模糊系統自我學習 (Self-learning) 或自我組織 (Self-organizing) 的方向努力¹², 期望藉由各式各樣人性化的自學方案, 讓模糊推論系統具備學習能力, 達到自我建立知識庫的目的, 免去人工建立之苦, 並且希望經由此種方式所架構出來的模糊推論系統能超越人工建立者, 甚至超越原來的領域專家。

(2) 使用者介面:

使用者介面提供了使用者與系統互動的管道, 使用者可以向系統提出問題, 系統則在知識庫中尋求最接近的解答回復給使用者。

(二) 類神經網路與機器認知

類神經網路 (Neural networks) 又稱人工神經網路 (Artificial neural networks) 或簡稱神經網路。顧名思義, 是以人工的網路架構來模擬人腦之神經組織, 並希望透過有效的學習法則, 使其能像人腦一樣, 具備學習能力。類神經網路的研究起源極早, 一般認定 1940 年代為其「萌芽期」。心理學家 McCulloch 和數學家 Pitts 於 1943 年共同提出神經元的數學模式, 並取兩人姓氏開頭第一個字母命名

¹² 林信成、陳永耀, 「自我學習法則在模糊推論系統上之應用」, 模糊系統學刊, 第一卷, 頁 7-26, 民 84 (Journal of Chinese Fuzzy Systems Association, Vol.1, pp7-26, 1995)。

為 MP 模型，開創神經網路研究之門。¹³之後有許多學者紛紛加入研究行列，如 1949 年 Hebb 提出著名的赫比學習法則 (Hebbian learning rule)¹⁴，至今仍是極重要的神經網路學習法之一。1954 年 Minsky 完成首部類神經電腦 (Neurocomputer)¹⁵。1958 年 Rosenblatt 提出感知器 (Perceptron) 模型¹⁶，首次以臨界值單元 (Threshold elements) 模擬人腦神經元，並透過訓練及改變鍵結方式達到分類 (Classification) 功能。1960 年代初期，Widrow 提出 ADALINE，並發表著名的 Widrow-Hoff 學習法則¹⁷，該法則可使訓練誤差的平方和最小。然而正當各家學說蓬勃發展之際，研究人工智慧的重量級大師 Minsky 和 Papert 卻於 1969 年出版了「感知器 (Perceptron)」一書¹⁸，對於感知器之類的網路 (Perceptron-class networks)，悲觀地指出其侷限性，比如連簡單的 XOR 問題都無法解決。Minsky 的質疑一直到 1980 年代中期，仍得不到合理的解答，這段期間是類神經網路的黑暗期，不過仍有不少重要的成果陸續被發表。1980 年代是神經網路的茁壯期。首先，Hopfield 於 1982 年提出回饋網路模型¹⁹，引入能量函數的概念，成為聯想記憶體 (Associative memory) 之基礎。1986 年，突破性的研究由 McClelland 和 Rumelhart 兩位學者共同發表²⁰，以微分連鎖律解決了多層 (Multilayer) 神經網路的學習瓶頸，也一併解決 Minsky 的問題，這就是著名的多層神經網路「逆傳遞學習法」(Backpropagation learning rule)，簡稱 BP 學習法，它也是迄今為止最廣為使用的神經網路。自此，神經網路的研究如雨後春筍，形成一股熱潮。其應用也因此滲透到各個領域裡，尤其在分類、系統識別、影像處理、樣式辨認、智慧型控制、非線性最佳化等方面都有很好的成果。

神經網路的模型雖多，但大致可以區分如下：²¹

1. 以網路結構分類：
 - (1). 前饋網路 (Feedforward networks) : 以多層感知器 (Multilayer perceptron) 為代表；
 - (2). 回饋網路 (Feedback networks) : 以 Hopfield 網路為代表。
2. 以學習法則分類：
 - (1). 監督式學習法 (Supervisory learning) : 有訓練樣本可供學習依循者，如 BP 即是；
 - (2). 非監督式學習法 (Unsupervised learning) : 無訓練樣本者，如自組織神經網路屬之；

基本上，類神經網路是由許多基本神經元相互連接所組成。圖 4 為一個典型的三層前饋網路，每個神經元 (圖 4(a)) 都具有一定數量的輸入節點 (Input

¹³ W. S. McCulloch and E. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bull. Math. Biophys.* 5, pp. 115-133, 1943.

¹⁴ D. O. Hebb, *The organization of behavior, a neuropsychological theory*, New York: John Wiley, 1949.

¹⁵ M. Minsky, "Neural nets and brain," *Doctoral Dissertation*, Princeton Univ., NJ, 1954.

¹⁶ F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psych. Rev.* 65, pp. 368-408, 1958.

¹⁷ B. Widrow, "Generalization and information storage in networks of adaline 'neurons'," *Self-organizing Systems*, pp. 435-461, 1962.

¹⁸ M. Minsky and S. Papert, *Perceptrons*, Cambridge, Mass.: MIT Press, 1969.

¹⁹ J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci.* 79, pp. 2554-2558, 1982.

²⁰ T. L. McClelland and D. E. Rumelhart, *Parallel Distributed Processing*, Cambridge: MIT Press and the PDP Research Group, 1986.

²¹ J. M. Zurada, *Introduction to Artificial Neural Systems*, West Pub. Co., 1992.

nodes), 如同人腦無以數計的神經末梢, 可接受來自外界的刺激; 每個輸入節點有一個對應的權值 (Weight), 相當於腦神經的鍵結強度; 所有輸入經過加權後, 經由一個非線性的網路函數 (Net function) $f(\cdot)$ 輸出, 模擬人腦對感知的非線性現象; 每個神經元經由複雜的方式連結 (Connection), 如圖 4 (b), 就像人腦神經組織的互相糾結一般。因此, 基本上每個神經元都具有簡單的非線性處理能力, 當連接組合成網路之後即具備複雜的非線性映射 (Nonlinear mapping) 功能。1989 年 Hornik 等人證明了一個多層的前饋式神經網路可經由學習而達到任意的非線性函數映射。這裡所謂的「學習」, 其實只是透過網路參數或權值的調整, 達到降低誤差的目的。而所謂逆傳遞 (BP) 學習法, 顧名思義就是將誤差由輸出端經網路一路逆傳回輸入端, 以改變網路參數或權值。所使用的方法則是梯度法 (Gradient decent method), 其學習方程式使用微分連鎖律即可輕易求得²²。

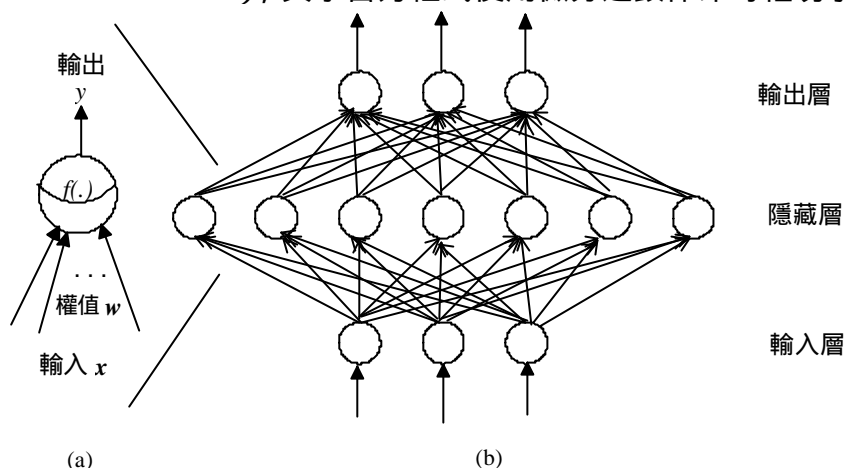


圖 4 典型的三層前饋式類神經網路

(三) 學習法則與機器學習

學習是有機體獲取知識的重要過程, 智慧型系統若要能自動累積知識, 也應具備模擬有機體能自我學習的能力。然而, 即使如人類一般的高等動物, 對於本身如何學習這件事, 也都還處於探索階段, 無法完全瞭解。因此, 以當前的科技而言, 在機器學習方面仍處於採用以較直覺簡單的方式所定義的學習法則供機器遵循的階段, 以下就兩種較常見的機器學習法加以說明。

1. 增強式學習法 (Reinforcement learning):

增強式學習法 (Reinforcement learning) 是根源自心理學大師帕弗洛夫 (Pavlov) 的制約實驗, 這個實驗讓狗把鈴聲和食物聯想在一起: 每次餵食前都先讓狗聽一段鈴聲, 久而久之, 只要鈴聲響起, 即使食物沒有出現, 狗也會因預期心裡而流涎三尺, 這就是有名的帕弗洛夫的制約理論 (Pavlovian conditioning theory)。

增強式學習法不同於監督式學習法及非監督式學習法之處在於: 監督式學習法有訓練樣本作為學習依據, 如同考題有標準答案一般, 答錯了馬上有答案供修正; 非監督式學習則完全沒有答案, 全靠自我摸索, 根本不曉得答案對不對, 直到「試」出來為止; 而增強式學習則介於兩者之間, 雖沒有標準答案, 至少有一些較粗略的訊息, 如「對、錯」或「好、壞」作為學習之依據, 學習動作就是根據這些訊息進行。

²² 同註20。

1983 年 Barto, Sutton 和 Anderson 三人首度將增強式學習法應用於自適應類神經單元 (Neuronlike adaptive elements), 解決困難的車與桿系統 (cart-pole system)²³ 平衡問題後, 始激起增強式學習法的研究熱潮²⁴。Barto 的理論脫胎自早期的 BOXES 系統, BOXES 方法是將系統的四個狀態變數 (車位置、車速度、桿角度以及桿角速度) 所構成的狀態空間分割成 162 個 boxes, 而每個 box 有一個對應的控制輸出, 當桿子倒下或車子撞倒邊界時, 這個失敗事件會產生一個失敗信號給決策/控制器, 學習過程就是根據這個失敗信號來對不當的決策動作進行獎懲²⁵。Barto 等人提出, BOXES 系統可以一個簡單的神經元取代, 這個神經元稱為關連搜尋單元 (ASE), 其中有一個解碼器是用來將狀態變數轉換成 162 位元的輸入變數 x_j , 其中只有一個位元為 1, 代表該狀態正好落在這個 box 內。而 ASE 的功用就是根據系統提供的增強信號 (或稱獎懲信號), 來調整該 box 的權值。當失敗事件未發生時, 獎懲信號為 0, 決策動作不受懲罰; 而當失敗事件發生時, 獎懲信號為 -1, 決策動作的權值因受懲罰而降低, 因此, 不當動作的影響力便減少, 是不是很人性化呢? 但是, 這種方式每次都要等到失敗事件發生了, 才調整動作一次, 導致學習速度緩慢。因此, Barto 等人很技巧的加入另一個神經單元, 稱為適應評估單元 (ACE), 用來「預測」失敗是否即將發生, 以便及早進行修正, ASE 的權值不再直接由失敗事件所產生的外部加強信號來調整, 而是由 ACE 所評估出來的內部加強信號來調整。這個內部加強信號不再只有 0 和 -1 兩個值, 而有正負之分, 愈正表示獎勵愈大, 愈負代表懲罰愈重, 因此, 學習過程便隨時隨刻都在修正。這就有點像狗聽到鈴聲, 預測到肉即將出現而流口水一樣。

2. 進化式學習法 (Evolution learning):

進化式學習法以「遺傳演算法」(Genetic algorithms, 簡稱 GA) 為代表, 乃是以達爾文的「進化論」為其發展根基, 為一種模擬「物競天擇」及「自然遺傳」的搜尋法則 (Search algorithm), 最初是由密西根 (Michigan) 大學的 John Holland 教授所提出的²⁶。其主要目的在於建立一個保有生物自然特性的人工系統, 以便於模擬及解釋自然生物系統的進化過程。R. B. Hollstien 和 K. A. De Jong 的研究指出遺傳演算法, 是一個極有效的函數最佳化法[39]。De Jong 在其後來的研究更指出遺傳演算法是一種強健搜尋技術 (Robust search technique) [40]。如今, 此一法則已廣泛地應用於自然科學及人工智慧等領域[38]。在工程應用方面, 遺傳演算法可用來尋找一個問題 (或函數) 的近似最佳解。應用遺傳演算法求解「最佳化問題」之前, 必須將所遭遇的問題轉化成對應的函數, 稱為適應函數 (Fitness function), 適應函數代表著系統對外在環境的適應能力 (Fitness), 相當於該系統的性能指標 (Performance index), 適應函數值愈大表示該系統的性能愈好, 反之, 表示性能愈差。遺傳演算法的目的便是透過一些擬生物化的人工運算過程, 如重生、交配、突變...等進行演化, 最後, 尋得適應函

²³ 車與桿平衡問題是：在一部能左右移動的台車上, 豎放一支可自由擺動的垂直桿, 想辦法移動台車使桿不致倒下, 而且台車不能撞倒邊界。這是一個模擬孩童將竹竿豎立在地上玩平衡遊戲的系統。

²⁴ A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. on Syst. Man and Cyber.*, vol. 13, no. 5, pp. 834-846, 1983.

²⁵ D. Michie and R. A. Chambers, "BOXES: An experiment in adaptive control," in *Machine Learning* 2, E. Dale and D. Michie, Eds. Edinburgh: Oliver and Boyd, pp. 137-152, 1968.

²⁶ J. H. Holland, "Genetic algorithms and the optimal allocations of trials," *SIAM Journal of Computing*, 2(2), 88-105 (1973).

數的最佳解。如今，有許多文獻提出各式各樣不同的方法來改良遺傳演算法，但其基本精神都是從「簡易遺傳演算法」(Simple genetic algorithm , SGA) 發展出來的。因此，以下僅就 SGA 的運作過程加以說明。

運用遺傳演算法的基本運算子如重生、交配、突變之前，必須先完成如下之準備工作：

(1). 定義適應函數 (Fitness Function)

適應函數是遺傳演算法的性能指標 (Performance index)，例如 $f(x_1, x_2) = x_1^2 + 5x_2$ ，其中 x_i 稱為函數 f 的參數。遺傳演算法的目的就是要找到使 f 函數值最大的參數值 (x_1, x_2) 。

(2). 決定編碼 (Coding) 與解碼 (Decoding) 方式

為了有效的搜尋參數空間，首先要確認每個參數的搜尋範圍，再將每個參數以固定長度的字串加以編碼，最簡單也最廣為使用的編碼方式是二進位編碼 (Binary Coding)。在二進位編碼方式中，每個參數都預先被轉換成 n 位元的二進位數字，因此，如果函數 f 的某個參數 x_i 的搜尋範圍界定於 $[a_i, b_i]$ 之間，那麼，以二進位編碼之後，將迫使實際的 x_i 值量化至 $a_i + k \frac{b_i - a_i}{2^n}$ 及 $a_i + (k + 1) \frac{b_i - a_i}{2^n}$ 之間，其中 k 是介於 0 到 2^n 間的任一整數。

(3). 產生位元字串 (Bit string)

將每一個參數 x_i 串起來構成一個參數向量 $x = (x_1, x_2, \dots, x_m)$ ，同時將每個參數所對應的二進位碼串成二進位字串，即

$$\begin{bmatrix} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{bmatrix}$$

其中共 n 個位元為 x_i 所對應的二進位碼。如此一來，對於有 m 個參數，每個參數用 n 個位元編碼的函數而言，我們便擁有長度為 $N = m \times n$ 的位元字串。因此上式的二進位字串 p 便可改寫成

$$\begin{bmatrix} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{bmatrix}$$

其中 $d_j \in \{0, 1\}, j = 1, 2, \dots, N$ 。

在遺傳演算法的術語中，將位元字串 p 稱為一個個體 (Individual)，位元字串中的一個位元 (bit) d_j 則相對於該個體中的一個遺傳基因 (Gene)。不同的基因 (參數值) 會產生不同的個體 (參數組合)，因此每個個體 (位元字串) 對應於參數空間中的一個解。遺傳演算法便是要利用各種擬生物化的過程，保留優良基因，淘汰不良基因，使整體性能指標隨著一代一代演化而逐代提高。

(4). 產生原始族群 (Initial population)

在啟動遺傳演算法之前，必須先隨機產生 S 個第零代的個體 (位元字串) $\{p_1(0), \dots, p_s(0)\}$ ，稱為原始族群 (Initial population)，族群大小 (Population size) S 則視問題的複雜度而定。一般而言，愈複雜的問題需要愈大的族群規模來解決。由於每個個體代表一個解，因此， S 個原始個體便代表 S 個初始解。這 S 個初始解的性能指標可能都很低 (非最佳解)，遺傳演算法便是希望藉由以下即將提到的基本運算子，經過幾代演化之後，能一代比一代更好，使整體性能指標提高，最後達到最大值，求得適應函數的最佳解。

產生了原始族群之後，即可啟動遺傳演算法，演化過程如圖 5 所示，包含三個基本運算子 (Operators)：重生 (Reproduction)、交配 (Crossover) 及突變 (Mutation)。這些運算子作用於舊一代 (Old generation) 族群，產生新一代 (New generation) 族群。詳述如下：

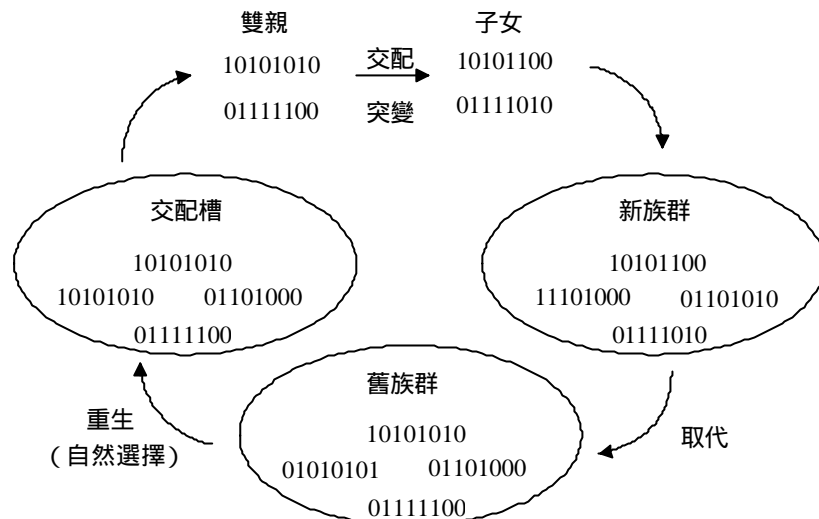


圖 5 遺傳演算法演化過程

(1). 重生 (Reproduction):

類似生物的無性生殖。根據每個個體的適應函數值高低，決定該個體被複製的機率。因此，性能指標高的個體，就會有較高的機率被選擇到而「自我複製」出下一代的新個體，無可置疑的，這是一個人工版的自然選擇 (Natural selection) 過程。因為性能指標差的個體，被選中自我重生的機率比性能指標高者小，以至於在新的族群中，性能差的個體會比舊族群中少，取而代之的是性能較好的個體。如果在舊族群中有 S 個個體，那麼在重生階段，自然選擇機制也必須複製出相同數目的新個體。這 S 個自我重生的個體將悉數被放入一個稱之為交配槽 (Mating pool) 的緩衝區中，等待進一步的繁衍。

有許多方法可以用來實現重生過程中的自然選擇機制，其中最簡易也最廣為採用的是「輪盤法」。此一輪盤不同於一般的等分格輪盤，其主要特色是輪盤中每個槽 (Slot) 的大小都是根據每個個體性能指標 (或適應能力, Fitness) 的百分比來設定的，也就是性能指標愈高者所佔據的盤面比例愈大。其分格比例大小可以下式求得

$$PS_i = \frac{f_i}{\sum_{i=1}^S f_i}$$

其中 f_i 為第 i 個個體的適應函數值， PS_i 可視為第 i 個個體被選中的機率。

自然選擇過程不斷反覆進行，每選中一個個體，即將其複製重生，放入交配槽中，直到交配槽中擁有 S 個個體為止。理論上，擁有較高適應能力的個體，將擁有較高的機率被複製，以至於能提供較多副本到交配槽中。此步驟完成後，由於交配槽中的個體僅是原個體的副本，所以尚未有異於原個體的新個體產生。

(2). 交配 (Crossover):

交配運算子提供了一個訊息 (Information) 交換的機制，使得族群中不同個體可經由隨機交配過程，互換基因，以產生新的子代 (Offspring)。首先，從交配槽中任挑兩者，稱之為雙親 (Parents)，然後隨機地在雙親字串的 N 個基因 (位元) 中任挑一點，稱之為交換點 (Crossover point)，再將位於交換點右側的雙親基因互換，生成兩個新的個體，如此即完成交配過程。因此所謂交配，其實只是將雙親的基因 (位元變數) 互換罷了。

交換點的產生亦是以亂數產生器完成，每次產生的亂數都是介於 $1 \sim N$ 之

間的整數。例如，假設 8 位元的雙親字串為

$$p_1 = 10101010$$

$$p_2 = 01111100$$

若隨機產生之交換點為 5，亦即將雙親由左算起第 5 位元右側的所有位元（畫有底線者）互換，則產生的新個體為

$$p_1' = 10101100$$

$$p_2' = 01111010$$

其中， p_1' 之交換點左側 5 位元的基因來自 p_1 （父親），而右側 3 位元的基因來自 p_2 （母親）；同理 p_2' 有 5 個基因來自母親，3 個基因來自父親。由以上的簡單例子可知，子代的兩個新個體分別含有親代雙親的部份基因樣式（Pattern）。因此，交配步驟完成後，基本上已有異於舊個體（親代）的新個體（子代）產生，但基因樣式仍沿襲自雙親，並無新的基因樣式產生，有賴突變來產生新基因樣式。

另外，並非所有被挑中的雙親皆能進行交配，而是根據預設的交配機率（Probability of crossover）來決定交配發生的頻率。交配機率一般設為 0.8 ~ 1）。例如：若交配機率設為 0.8，則每次從交配槽中挑出兩個雙親個體時，必須先以擲骰子的方式決定是否進行交配，亦即先產生一個介於 0 ~ 1 之間的實數亂數，若該數小於 0.8 則進行交配產生新字串；否則只要將雙親基因完全複製給子孫即可，而不進行交配。

(3). 突變（Mutation）：

突變過程是將交配後產生的子代，根據預設的突變機率進行突變。其作法只是將隨機選定的位元反轉（0 變 1，1 變 0）罷了。一般而言，突變機率都設定得不高（約 0.001 左右），突變可以引進新的基因樣式，避免過早收斂（Premature convergence）。而且，突變本身是一個在參數空間中亂走（Random walk）的隨機過程，可以開發新的搜尋領域，防止收斂於局部最佳點（Local optimum），而探尋整體最佳點（Global optimum），但又由於突變機率通常不高，故不會使遺傳演算法流於完全的「亂走」演算。

經過以上三個繁衍過程，舊一代（Old generation）族群的 S 個舊個體便衍生出新一代（New generation）族群的 S 個新個體。這個新族群必須取代舊族群，並負起繼續繁衍下一代的責任。

由於演化過程是根據性能指標為依據，因此理論上子代的性能指標應高於親代。如此一代一代演化下去，終能找到適應函數的近似最佳解。

四、整合軟性計算技術之智慧型控制與決策系統

控制（Control）與決策（Decision）是人類極重要的心智活動，其運作過程是根據所感知的條件，進行推論、判斷，最後採取正確的行動。既然類神經網路具備類似人腦的認知能力，模糊推論可用來模擬人腦的思考邏輯，而學習法則又可有效達到機器學習的目的，自然而然地，許多軟性計算的研究乃朝著結合彼此優點的方向努力，以建構具備更高機器智商的控制與決策系統。雖然目前已有為數眾多的結合式模型被發表出來，但總歸起來不外乎「以類神經網路之軀殼，包裹模糊推論之思想，再配以學習法則為靈魂，達到更符合人類認知結構與思維模式的目的」。

若要追溯類神經模糊系統（Neurofuzzy systems）的研究，最早可能是 1974

年²⁷，S. C Lee 和 E. T. Lee 兩位學者，將 MP 模型的二值輸出一般化，使其成為介於 [0, 1] 之間的多值邏輯。1985 年，Keller 和 Hunt 兩人首度提出將模糊歸屬函數概念加入感知器²⁸。此後陸續的發展並未普遍，直到 1990 年代才開始蓬勃起來。Takagi 於 1990 年所發表的論文中²⁹，分別對模糊理論和神經網路的融合及未來可能的發展方向，做了一番回顧及探討。類神經模糊系統較為完整的架構則於 1991 年由 C. T. Lin 及 C. S. G. Lee 共同提出³⁰，該研究使用五層 (five layers) 的前饋式類神經網路，實現模糊推論系統的四大功能模組，並利用逆傳遞學習法調整歸屬函數，稱為神經網路式模糊邏輯控制決策系統 (Neural-network-based fuzzy logic control and decision systems)，簡稱 NN-FLC，略述其功能如下：第一層為輸入節點 (Input linguistic nodes)，主要作用只是將輸入值直接傳至下一層，可視為一個單點模糊化模組 (Singleton fuzzifier)；第二層為輸入語意項節點 (Input term nodes)，每個節點的動作函數 (Activation function) 對應於一個歸屬函數，因此，這一層可視為定義輸入歸屬函數的資料庫，從而進行輸入語意值之真值 (Truth value) 運算；第三層為規則節點 (Rule nodes)，是知識庫的核心所在，主要用來記錄規則及進行規則前提部之推論運算，其節點動作函數為模糊 AND 運算，因此，這一層的每個節點對應一條規則，其輸出值則對應該條規則的觸發強度；第四層為輸出語意項節點 (Output term nodes)，主要是對觸發規則進行模糊 OR 運算；第五層為輸出節點 (Output linguistic nodes)，不但記錄著輸出歸屬函數，且負有解模糊化的任務。Lin 和 Lee 同時還依據逆傳遞法提出混合學習法 (Hybrid learning algorithm)，以訓練 NN-FLC 網路中的參數 (即歸屬函數)，然而先決條件是必須有訓練樣本，所以這是一個監督式學習架構。隨後，在 1993 年 Jang 也提出了一個借用神經網路逆傳遞法則的適應式模糊推論系統，稱為 ANFIS (Adaptive Network-based Fuzzy Inference System)³¹。在 ANFIS 中，第一層包括模糊化及歸屬函數的定義，負責將輸入轉換成適當的歸屬度，所必須調整的就是歸屬函數的參數。第二層負責規則前提部之 AND 運算，求取規則的觸發強度。第三層的任務是將規則觸發強度正規化 (Normalize)，相當於計算模糊基底函數 (Fuzzy basis functions)。第四層主要工作為計算個別規則的輸出值，而 ANFIS 的規則輸出是採用 Takagi-Sugeno 模型的多項式形式³²，因此所要學習的參數就是多項式的係數。第五層則只負責將這些輸出作總和的工作。有別於一般逆傳遞學習法的是 ANFIS 所採用的是時間式逆傳遞法 (Temporal backpropagation, TBP)，TBP 的觀念最早是由 Widrow 所提出，是先將時間分段，再將每一時段的系統組態視為單一節點，眾多時段節點便可串成一個複雜的時間網路。每個時段的輸出狀態再與預期軌跡進行誤差量測，然後將此誤差經由時間網路逆傳以調整網路參數，即可達到學習目的。比較 NN-FLC 與 ANFIS 可以發現其架構有許多相通之處，可見這是類神經模糊系統最常使用的模式。另外，有

²⁷ S. C. Lee and E. T. Lee, "Fuzzy sets and neural networks," *J. Cyber.*, vol. 4, pp. 83-103, 1974.

²⁸ J. M. Keller and D. Hunt, "Incorporating fuzzy membership functions into the perceptron," *IEEE Trans. on Pattern Anal. Mach. Intell.* vol. 7, pp. 693-699, 1985.

²⁹ H. Takagi, "Fusion technology of fuzzy theory and neural networks - survey and future directions," *Proc. Int. Conf. on Fuzzy Logic and Neural Networks*, Japan, pp. 13-26, 1990.

³⁰ C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computer*, vol. 40, no. 12, pp.1320-1336, 1991.

³¹ J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst. Man. Cyber.*, vol. 23, no. 3, pp. 665-685, 1993.

³² T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on Syst. and Cyber.*, Vol. SMC-15, No. 1, pp. 116-132, 1985.

些研究使用輻徑基底函數網路 (Radial basis function networks) 與模糊推論系統相融合³³。更有許多研究顯示：模糊推論系統與前饋式多層神經網路，從數學觀點看來是完全等效的，都可以當作通用的函數近似器 (Universal function approximator)^{34, 35}，這使得軟性計算理論的整合更向前邁出一大步。

增強式學習法利用獎勵、懲罰的觀念提供極具人性化的學習機制，自從由 Barto 等人的 ACE/ASE 架構實現後，許多研究便朝向將其與其他軟性計算技術整合的方向努力。其中，最有名的是 C. C. Lee 將類似的架構應用至模糊推論系統，以解決同樣的車與桿平衡問題，³⁶其架構雖然比原來 Barto 的複雜許多，但觀念其實是相通的。兩者最大的差異在於 Lee 採用模糊推論作為決策及控制中樞；其次，解碼器由一個模糊解碼器取代，以便做重疊性的狀態空間分割；再者，以關連評估單元 ACN (Associative critic neuron) 和關連學習單元 ALN (Associative learning neuron) 取代 ACE 和 ASE，只不過 ALN 所學習的是模糊知識庫而非神經元的權值。事實上，Lee 的前提部歸屬函數是固定的，規則數也是固定的，唯一要學習的只是結論部的歸屬函數參數，並且 Lee 採行了簡化作法，只用三角形歸屬函數並僅調整其頂點。ACN 同樣負責預測及評估決策動作的成效，並送出評估的訊息。ALN 則根據 ACN 送來的評估報告，調整知識庫各輸出歸屬函數的頂點，其實仍是利用簡單的獎懲原理，以訓練決策中樞採取適當的動作。要注意的是，在 ACE/ASE 架構中，ASE 的輸出直接就代表決策結果；而在 ACN/ALN 架構中，ALN 的輸出代表的是歸屬函數，真正的決策輸出必須經由所有規則的合成才能得到。Berenji 和 Khedkar 則在 1992 年提出 GARIC (General approximate reasoning-based intelligent controller) 架構³⁷，基本上和前述的架構有些類似，主要也是分為兩大部份：動作評估網路 AEN (Action evaluation networks) 和動作選取網路 ASN (Action selection network)。不同的是不再使用單一神經元作為評估器/控制器，而採用較複雜的網路型態。決策中樞的 ASN 用了一個五層的類神經網路進行模糊推論，而評估用的 AEN 則為一個兩層的類神經網路。另外，值得一提的是在 GARIC 中有一個隨機動作修正器 (Stochastic action modified)，這個修正器根據 ASN 所送來的建議決策，及 AEN 評估出來的內部增強信號，隨機產生一個高斯分佈的決策信號。如果決策愈好，評估器的增強信號愈大，則標準差較小，表示此一決策較被肯定；反之，較差的決策會得到較大的標準差，所以，較差的決策就會有較高的機率不再被重複。1994 年 Lin 和 Lee 所提出的架構則更複雜³⁸。他們用了兩個前節所提的 NN-FLC，構成決策器和評估器，而每個 NN-FLC 則分別由五層的神經網路組成模糊推論系統。Lin 和 Lee 的評估器稱為模糊預測器 (Fuzzy predictor)，它的前兩層 (輸入層和語意項層) 和動作網路的前兩層是共用的。但兩者的後三層 (規則、推論、解模糊) 則是獨立的。比較特殊的是，Lin 和 Lee 的學習機構，並不只有調整歸

³³ J. Nie and D. A. Linkens, "Learning control using fuzzified self-organizing radial basis function network," *IEEE Trans. on Fuzzy Syst.*, vol. 1, no. 4, pp. 280-287, 1993.

³⁴ J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. on Neural Net.*, vol. 4, no. 1, pp. 156-159, 1993.

³⁵ L. X. Wang, *Adaptive Fuzzy Systems and Control*, Englewood Cliffs, NJ: Prentice Hall, 1994.

³⁶ C. C. Lee, "A self-learning rule-based controller employing approximation reasoning and neural net concepts," *Int. J. of Intel. Syst.*, vol. 6, pp. 71-93, 1991.

³⁷ H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements", *IEEE Trans. on Neural Net.*, vol. 3, no. 5, pp. 724-740, 1992.

³⁸ C. T. Lin and C. S. G. Lee, "Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems", *IEEE Trans. on Fuzzy Syst.*, vol. 2, no. 1, pp.46-63, 1994.

屬函數的參數，還具有結構學習 (Structure learning) 的能力，能夠決定究竟需不需要增加網路節點。

遺傳演算法是軟性計算領域中的後起之秀，發展雖較晚，卻潛力十足。首度將遺傳演算法應用於模糊推論系統的學者是 Karr³⁹，Kristinsson 和 Dumont 兩位學者則將遺傳演算法應用於動態系統的識別與控制上⁴⁰。此後，大部份研究者大都是以遺傳演算法來達到自動學習、演化、建立模糊知識庫的目的，並盡量找尋出使得模糊系統性能最佳化的參數組合。以遺傳演算法學習模糊知識庫，大致可分為兩大類型：

其一，是固定模糊規則，搜尋最佳歸屬函數組合。此種作法要解決的兩個主要問題是：(1) 要如何將問題的任一種可能解，編碼成固定長度字串？(2) 要如何衡量每個字串 (個體) 的性能？對於第一個問題，最簡單的作法便是將規則庫中所有歸屬函數的參數，串成一個單一的參數向量再以前節所述的方法對其進行二進位編碼，如此每一個個體 (位元字串) 便代表一個獨立的模糊推論系統。至於第二個問題，最直接的作法便是以系統實際輸出值和預定輸出值的誤差平方和 (Summation of squared-error) 做為代價函數 (Cost function)，再取其倒數 (或以一極大常數相減) 當作適應函數，則誤差平方和最小化的問題便自然轉換成適應函數最大化的問題。當然，定義適應函數的方法有千百種，並非一成不變的，完全要根據問題的種類配合設計的要求，才能定義出恰當的適應函數。

其二，是固定歸屬函數找尋適當的模糊規則。如果歸屬函數的形狀及參數都已知，但卻缺乏模糊規則，此時便需要利用一些技巧，將模糊規則庫轉換至參數空間，再應用遺傳演算法來找尋適當的規則庫。最簡單的作法，是先將模糊語意值 (模糊集合) 編號，當作參數，再將其隨機填入規則庫中，然後把每條規則中的模糊集合號碼串起來，構成一個參數向量，對此參數向量編碼成位元字串即可產生一個個體，每個位元字串代表不同的規則庫，往後所有遺傳運算子便都作用其上。至於性能指標的定義則同前，不再贅述。

當然，若遭遇到又要尋找適當的規則，又要調整歸屬函數的複雜情況，則必須將以上兩者稍作修正再運用。

五、結論

電腦應用從簡單的資料處理演進到複雜的資訊管理，成功的改變了人們的工作和休閒的模式。隨著科技的演進，下一個世代的電腦必然朝向愈來愈智慧化的方向發展，因此，研究探討有關知識的自動擷取、處理和管理的理論和技術，乃是提昇機器智商的首要工作。如今，軟性計算理論的發展，配合知識工程領域的各種研究成果，為機器的擬人思考、認知學習，跨出了可喜的第一步。但這僅僅是一小步，離真正的目標還極遙遠，需要再努力的還非常多。

當然，智慧型系統的自我學習，並不僅限於本文所提的數種方法，另外還有各式各樣的適應式 (Adaptive) 自調式 (Self-tuning) 自組式 (Self-organizing) ... 等等方法，不一而足，文中所舉僅是犖犖大者。本文旨在提供讀者一個概觀性的導讀及新的思考方向，並未牽涉到非常深入的理論探討，希望藉此引起讀者興趣，進而作更深入的研究。對本文介紹內容有興趣的讀者，若欲瞭解更詳細的理論架構，則需參閱附註中的參考文獻原文。

³⁹ C. L. Karr, "Applying genetics to fuzzy logic," *AI Expert*, Mar. 1991, pp.38-43.

⁴⁰ K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. on Syst. Man and Cyber.*, vol. 22, no. 5, Sep./Oct. 1992, pp.1033-145.